G-002

自己組織化マップを用いたアミノ酸配列の解析
The Analysis of Amino Acid Sequences using Self Organizing Maps

高橋 剛志† 堂薗 浩†
Takeshi Takahashi Hiroshi Dozono

1. はじめに

2003 年にヒトゲノム計画が完了して以来，バイオインフォマティクスはますます活発になってきている．その中でもタンパク質の機能推定は重要な課題となっている．タンパク質の機能推定方法として，タンパク質の一次構造であるアミノ酸配列を一対一で比較する BLAST が有名であるが，多数のアミノ酸配列同士を比較する目的には使用しづらい．自己組織化マップは Kohonen が考案したニューラルネットワークの一種で，クラスタリング，分類，多数の多次元データの可視化などに使われている[1]．そこで，本研究では自己組織化マップを用いてアミノ酸配列やその断片を比較，分類し，タンパク質グループ間の関係の可視化を試みた．

1.1 アミノ酸配列

ヒトをはじめとするさまざまな生物は，生体内での情報伝達物質や摂取した食物を分解するための酵素としてさまざまなタンパク質を使用している．このようなタンパク質は 20 種類のアミノ酸によって構成されており，必要に応じて生体内で合成される．

生体内でタンパク質が合成される過程は，DNA 内の遺伝子が RNA に転写され，RNA が三塩基ごとにアミノ酸に翻訳される．翻訳されたアミノ酸は一列に並べられたアミノ酸配列を形成し，アミノ酸配列が折りたたまれてタンパク質となる．

ヒトゲノム計画によって決定された遺伝子から翻訳されるアミノ酸配列の中には，どのような三次元構造のタンパク質に折りたたまれるのかが未知なものや，どのような働きがあるのかが未知なものがある．

本研究では自己組織化マップを用いて，さまざまなアミノ酸配列群の比較を行った．

同様の研究として，生物種間の DNA 配列の比較を行ったものがあるが，この研究では短い DNA 配列の発生頻度を入力として学習を行っている[2]．本研究では配列そのものを入力ベクトルとして学習を行うことで，より細かい解析が行えるのではないかと考えた[3]．

本研究で使用したアミノ酸配列データは，インターネットを利用して京都遺伝子ゲノム百科事典：KEGG から取得した．

2. 自己組織化マップ

自己組織化マップは Kohonen が提案した教師なし学習を行うニューラルネットワークの 1 種である．一般的な自己組織化マップは入力層と出力層の二層で構成され，多量の入力データをクラスタリングする，高次元のデータを二次元上に写像する，といった目的で使用されている．

†佐賀大学大学院工学系研究科

入力データが持つさまざまな特徴をベクトルとして表し，このベクトルを入力ベクトルとする．

3. アミノ酸配列解析への適用

アミノ酸配列を自己組織化マップに学習させるために，アミノ酸配列から入力ベクトルを作成する必要がある．

本研究では，アミノ酸配列の一部分を切り出したプローブを入力ベクトルとした．長さ L のアミノ酸配列から長さ l のプローブを作成する場合，プローブは $L-l+1$ 個作成される．

参照ベクトルは，プローブ p の先頭から k 番目のアミノ酸が $p(k)$ である確率 $P(k,p(k))$ を $20l$ 個並べたものとする．

入力ベクトルと出力層の位置 (i,j) にある出力ノードの参照ベクトルとの距離 d は以下のように定義する．

$$d = l - \sum_{k=1}^{l} P_{ij}(k, p(k))$$

一般的な自己組織化マップでは逐次型学習法を用いてマップを生成することが多い．しかし，逐次型学習法では入力ベクトルを学習する順番によって最終的な学習結果のマップが変わってしまうことがある．そこで，本研究ではバッチ型学習法によってマップを生成する．

アミノ酸配列の先頭から n アミノ酸移動した位置から作成したプローブ p_n と，アミノ酸配列の先頭から $n+1$ アミノ酸移動した位置から作成したプローブ p_{n+1} は，アミノ酸配列上の 1 だけずれた位置から作成されたものであるため，出力層の隣接したノードに配置されるのが適切であると考えられる．そこで，p_n の勝利者ノードの 8 近傍から p_{n+1} の勝利者ノードを探索し，バッチ型のユニット更新時に，平均場近似を用いたシミュレーテッドアニーリングを使用して参照ベクトルを更新する．

本研究では以下の手順に従って自己組織化マップのバッチ型学習を行った．

(1) 初期化

すべての学習対称のアミノ酸配列から入力ベクトルとなる長さ l のプローブを作成し，すべての参照ベクトルが持つ確率 $P_{ij}(k,m)$ を

$$\sum_{m} P_{ij}(k,m) = 1$$

となるように初期化する．

(2) 学習フェーズ

(2.1) 各出力ノードが持つカウンタ $C_{ij}(k,m)$ をすべて 0 にする．

(2.2) プローブ p_t との距離 d が最小となる参照ベクトルを持つ暫定勝利者ノード W'_{ij} を探索する．ただし，アミノ酸配列の先頭から作成されたプローブ p_0 を入力ベクトルとする場合と，プローブ p_t のひとつ前のプローブ p_{t-1} に対する勝利者ノード

W_{ij} が見つからなかった場合にはすべての出力ノードから暫定勝利者ノード W'_{ij} を探索し，プローブ p_{t-1} に対する勝利者ノードが見つかった場合には勝利者ノードの周囲 8 近傍の出力ノードから暫定勝利者ノード W'_{ij} を探索する．

(2.3) すべての出力ノードから暫定勝利者ノード W'_{ij} を探索した場合には距離 d が閾値 TH_1 未満であれば暫定勝利者ノード W'_{ij} を勝利者ノード W_{ij} とし，そうでなければ勝利者ノードはなしとする．周囲 8 近傍の出力ノードから暫定勝利者ノード W'_{ij} を探索した場合には距離 d が閾値 TH_2 未満であれば暫定勝利者ノード W'_{ij} を勝利者ノード W_{ij} とし，そうでなければ勝利者ノードはなし，$p_{t+1} = p_t$ として手順(2.2)に戻る．

(2.4) 勝利者ノード W_{ij} が持つカウンタ $C_{ij}(k, p_t(k))$ を $1 \le k \le l$ についてインクリメントする．

(2.5) 未入力のプローブがある場合はプローブ $p_{t+1} = p_{n+1}$ として手順(2.2)に戻る．未入力のプローブがない場合には更新フェーズに進む．

(3) 更新フェーズ

(3.1) シミュレーテッドアニーリングを行う温度 $TEMP$ を設定する．

(3.2) すべての出力ノード U_{ij} に対して以下の処理を行う．

(3.2.1) すべての k，m について，計算用のカウンタ $C'_{ij}(k,m)$ にカウンタ $C_{ij}(k,m)$ の値を代入する．

(3.2.2) 出力ノード U_{ij} の近傍にあるすべての近傍ノード U_{xy} に対して以下の処理を行う．

(3.2.2.1) 下記の式で定義される距離 $dist$ が最大となる整数 SN を $-l/2 \le SN \le l/2$ で求める．

$$dist = \frac{\sum_k \sum_m P_{ij}(k,m) \times P_{xy}(k+SN,m)}{l - |SN|}$$

(3.2.2.2) すべての k，m について，計算用のカウンタ $C'_{ij}(k,m)$ に $fn(d_u) \times P_{xy}(k+SN,m)$ を加算する．ただし，d_u は出力ノード U_{ij} と近傍ノード U_{xy} の間のユークリッド距離で，

$$d_u = \sqrt{(x-i)^2 + (y-j)^2}$$

であり，$fn(d_u)$ は近傍関数である．

(3.3) 平均場近似を用いたシミュレーテッドアニーリングによって $P_{ij}(k,m)$ を更新する．

$$P_{ij}(k,m) = \frac{\exp\left(\left(C'_{ijMAX}(k) - C_{ij}(k,m)\right)/TEMP\right)}{\sum_m \exp\left(\left(C'_{ijMAX}(k) - C_{ij}(k,m)\right)/TEMP\right)}$$

ただし，

$$C'_{ijMAX}(k) = \max_m C'_{ij}(k,m)$$

である．

(3.4) 学習終了条件を満たしていれば終了し，そうでなければ手順(2)に戻る．

4. 解析結果

図１,図２に以上のアルゴリズムに従ってアミノ酸配列を解析した結果を示す．図１は,複数の生物が持つアミノ酸代謝に関与するアミノ酸配列群を入力ベクトルとして与え,各プローブに対する勝利者ノードをプローブの属する生物によって色付けしたものである．図２は,ヒトが持つ代謝系に関与するアミノ酸配列群を入力ベクトルとして与え，各プローブに対する勝利者ノードをプローブの属する代謝系によって色付けしたものである．このとき学習に使用したパラメータはどちらも同じもので,マップサイズは 30×30,プローブ長は 5,学習終了条件は手順(2)(3)を 50 回繰り返すこと,とした．

複数の生物が持つ同一機能のアミノ酸配列群を入力ベクトルとした場合は,勝利者ノードが群ごとに隣接していることが分かる．

また,ヒトが持つ代謝系に関与するアミノ酸配列群を入力ベクトルとした場合は,勝利者ノードが群ごとに隣接していることが分かる．

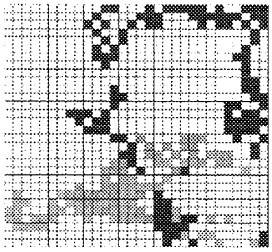

図1．生物種による分類

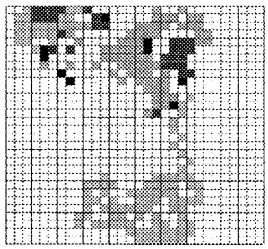

図2．機能による分類

5. まとめ

本研究では,平均場近似を用いたシミュレーテッドアニーリングを使用したバッチ型自己組織化マップによって多数のアミノ酸配列群の分類を行った．

複数の生物が持つ同一機能のアミノ酸配列群を入力ベクトルとした場合，ヒトが持つ代謝系に関与するアミノ酸配列群を入力ベクトルとした場合のどちらも群ごとに分類された．この結果を生物学的に解析することで，生物種間や代謝系間の進化的関係を見ることができると考えられる．

今後の課題としては,学習に使用したアミノ酸配列群の内から１つのアミノ酸配列をマッピングした場合のマッピングのされ方を確認すること,学習に使用していないアミノ酸配列をマッピングした場合のマッピングのされ方を確認することが挙げられる．

また,自己組織化マップでマップの各ユニットにおけるプローブの立体構造や機能を学習することで，立体構造や機能が未知のアミノ酸配列の立体構造や機能を推定することが出来ると考えられる．

参考文献

[1] Teuvo Kohonen：“自己組織化マップ　改訂版”，シュプリンガー・フェアラーク東京株式会社，2005

[2] Abe,T., Ilemura, T., S. Kinouch, S. and Sugawara, H.：“A Novel Bioinformatics Strategy for Phylogenetic Study of Genomic Sequence Fragments:Self Organizing Map(SOM) of Oligonucleitide Frequencies”, Proceedings of 5th Workshop on Self Organizing Maps, pp.669--676(005)

[3] Hiroshi Dozono：“An Algorithm of SOM using Simulated Annealing in the Batch Update Phase for Sequence Analysis”, Proceedings of 5th Workshop on Self Organizing Maps, pp.171--178(005)